

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used **string macro transform replace**

Found **581** of **139,567**

Sort results
by

Display
results

☒ [Save results to a Binder](#)

☐ [Search Tips](#)

☐ Open results in a new
window

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [MACRO: a programming language](#)

Stephen R. Greenwood

December 1979 **ACM SIGPLAN Notices**, Volume 14 Issue 12

Full text available: pdf(1.41 MB)

Additional Information: [full citation](#), [references](#), [citations](#)

2 [Macro processing in high-level languages](#)

Alexander Sakharov

November 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 11

Full text available: pdf(709.71 KB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

A macro language is proposed. It enables macro processing in high-level programming languages. Macro definitions in this language refer to the grammars of the respective programming languages. These macros introduce new constructs in programming languages. It is described how to automatically generate macro processors from macro definitions and programming language grammars written in the lex-yacc format. Examples of extending high-level languages by means of macros are given.

3 [Session 1 \(full technical papers\): evolution in source code: Challenges of refactoring C programs](#)

Alejandra Garrido, Ralph Johnson

May 2002 **Proceedings of the international workshop on Principles of software evolution**

Full text available: pdf(687.83 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Refactoring has become a well-known technique for transforming code in a way that preserves behavior. Refactorings may be applied manually, although manual code manipulation is error prone and cumbersome, so maintainers need tools to make automatic refactorings. There is currently extensive literature on refactoring object-oriented programs and some very good tools for refactoring Smalltalk and Java code. Although there is more code written in C or C++ than in any other language, refactoring too ...

Keywords: C programming, preprocessor directives, refactoring

4 [Stack Machines and Classes of Nonnested Macro Languages](#)

Joost Engelfriet, Erik Meineche Schmidt, Jan van Leeuwen

5 Tool support for refactoring functional programs

Huiqing Li, Claus Reinke, Simon Thompson

August 2003 **Proceedings of the ACM SIGPLAN workshop on Haskell**

Full text available:  pdf(156.41 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Refactorings are source-to-source program transformations which change program structure and organisation, but not program functionality. Documented in catalogues and supported by tools, refactoring provides the means to adapt and improve the design of existing code, and has thus enabled the trend towards modern agile software development processes. Refactoring has taken a prominent place in software development and maintenance, but most of this recent success has taken place in the OO and XP co ...

Keywords: Haskell, language-aware programming environments, program transformation, refactoring, semantic editors

6 BIGMAC II: A FORTRAN language augmentation tool

Eugene W. Myers, Leon J. Osterweil

March 1981 **Proceedings of the 5th international conference on Software engineering**

Full text available:  pdf(1.00 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the motivation, design, implementation, and some preliminary performance characteristics of BIGMAC II, a macro definition capability for creating language enhancers and translators. BIGMAC II enables the user to specify transformations through STREX, a FORTRAN-like language, which enables the specification of macros which are then used to interpretively alter incoming programs. BIGMAC II is specially adapted to the processing of FORTRAN programs. This paper shows how it ...

7 Migration of legacy web applications to enterprise Java™ environments net.data® to JSP™ transformation

Yu Ping, Jianguo Lu, Terence C. Lau, Kostas Kontogiannis, Tack Tong, Bo Yi

October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies conference on Collaborative research**

Full text available:  pdf(165.69 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


As Web technologies advance, the porting and adaptation of existing Web applications to take advantage of the advancement has become an issue of increasing importance. Examples of such technology advancement include extensible architectural designs, more efficient caching protocols, and provision for customizable dynamic content delivery. This paper presents an experience report on the migration of legacy IBM® Net.Data® based applications to new enterprise Java

Keywords: Java 2 Enterprise Edition (J2EE™), JavaBeans, JavaServer pages, Net.Data, SQL, migration, model-view-controller (MVC), transformation

8 Automating software analysis and testing using a program transformation system

G. Kotik, L. Markosian

November 1989 **ACM SIGSOFT Software Engineering Notes , Proceedings of the ACM SIGSOFT '89 third symposium on Software testing, analysis, and verification**, Volume 14 Issue 8


Full text available:  pdf(888.08 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe an approach to software analysis and test generation that combines several technologies: object-oriented databases and parsers for capturing and representing software; pattern languages for writing program templates and querying and analyzing a database of software; and transformation rules for automatically generating test cases based on the analysis results, and for automatically creating program "mutants" to determine adequacy of coverage of the test cases. We pre ...

9 [Graphical search and replace](#)

David Kurlander, Eric A. Bier

June 1988 **ACM SIGGRAPH Computer Graphics , Proceedings of the 15th annual conference on Computer graphics and interactive techniques**, Volume 22 Issue 4

Full text available:  pdf(1.42 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Graphical search is a technique for finding all instances of a graphical pattern in a synthetic picture in which objects are regions bounded by lines and curves. The pattern may describe shape, color and other properties. Matched objects may be allowed to differ from the pattern in rotation and scale or may differ in shape by a specified tolerance. *Graphical replace* is a technique for replacing the shape, color, or other properties of matched objects with new properties described i ...

Keywords: curve matching, graphical editing, graphical grammars, graphical macros, search and replace

10 [Algorithm 803: a simpler macro processor](#)

William A. Ward

June 2000 **ACM Transactions on Mathematical Software (TOMS)**, Volume 26 Issue 2

Full text available:  pdf(73.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Macro processors have been in the computing tool chest since the late 1950's. Their use, though perhaps not what it was in the heyday of assembly language programming, is still widespread. In the past, producing a full-featured macro processor has required significant effort, similar to that required to implement the front-end to a compiler augmented by appropriate text substitution capabilities. The tool described here adopts a different approach. The text containing macro definitions and ...

Keywords: awk, portable, simple

11 [The Vienna Definition Language](#)

Peter Wegner

January 1972 **ACM Computing Surveys (CSUR)**, Volume 4 Issue 1

Full text available:  pdf(3.89 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 [Syntax macros and extended translation](#)

B. M. Leavenworth

November 1966 **Communications of the ACM**, Volume 9 Issue 11

Full text available:  pdf(628.14 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)


A translation approach is described which allows one to extend the syntax and semantics of a given high-level base language by the use of a new formalism called a syntax-macro.

Syntax-macros define string transformations based on syntactic elements of the base language. Two types of macros are discussed, and examples are given of their use. The conditional generation of macros based on options and alternatives recognized by the scan are also described.

13 Functional memory-based dynamic microprocessors for higher level languages

George E. Rossmann, Louise H. Jones

May 1973 **ACM SIGPLAN Notices , Proceedings of the meeting on SIGPLAN/SIGMICRO interface**, Volume 9 Issue 8

Full text available:  [pdf\(1.05 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Dynamic microprocessors have been proposed as a means for providing the basic instruction sets necessary for efficient processing of a variety of higher level languages with specific hardware. Functional memory-based dynamic processors consisting of a collection of identical modules formed from writable associative arrays offer a more general solution to this problem. The basic properties of functional memory modules are discussed; and the structure of a functional memory-based microprocess ...

14 The FINITE STRING Newsletter: Abstracts of current literature

Computational Linguistics Staff

January 1987 **Computational Linguistics**, Volume 13 Issue 1-2

Full text available:  [pdf\(6.15 MB\)](#) 


Additional Information: [full citation](#)

[Publisher Site](#)

15 A finite state and data-oriented method for grapheme to phoneme conversion

Gosse Bouma

April 2000 **Proceedings of the first conference on North American chapter of the Association for Computational Linguistics**


Full text available:  [pdf\(709.26 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

A finite-state method, based on leftmost longestmatch replacement, is presented for segmenting words into graphemes, and for converting graphemes into phonemes. A small set of hand-crafted conversion rules for Dutch achieves a phoneme accuracy of over 93%. The accuracy of the system is further improved by using transformation-based learning. The phoneme accuracy of the best system (using a large rule and a 'lazy' variant of Brill's algorithm), trained on only 40K words, reaches 99%.

16 A language independent macro processor

William M. Waite

July 1967 **Communications of the ACM**, Volume 10 Issue 7

Full text available:  [pdf\(1.06 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The problem of obtaining starting values for the Newton-Raphson calculation of \sqrt{x} on a digital computer is considered. It is shown that the conventionally used best uniform approximations to \sqrt{x} do not provide optimal starting values. The problem of obtaining optimal starting values is stated, and several basic results are proved. A table of optimal polynomial starting values is given.

17 Automatic transformation of series expressions into loops

Richard C. Waters

January 1991 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 13 Issue 1

Full text available:  pdf(3.36 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


The benefits of programming in a functional style are well known. In particular, algorithms that are expressed as compositions of functions operating on sequences/vectors/streams of data elements are easier to understand and modify than equivalent algorithms expressed as loops. Unfortunately, this kind of expression is not used anywhere near as often as it could be, for at least three reasons: (1) most programmers are less familiar with this kind of expression than with loops; (2) most pro ...

Keywords: sequences, series, streams, vectors

18 Programmable syntax macros

Daniel Weise, Roger Crew

June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation**, Volume 28 Issue 6

Full text available:  pdf(1.09 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Lisp has shown that a programmable syntax macro system acts as an adjunct to the compiler that gives the programmer important and powerful abstraction facilities not provided by the language. Unlike simple token substitution macros, such as are provided by CPP (the C preprocessor), syntax macros operate on Abstract Syntax Trees (ASTs). Programmable syntax macro systems have not yet been developed for syntactically rich languages such as C because rich concrete syntax requires the manual con ...

19 REGULUS: A general purpose macro processor based on regular expressions part one

Harvey Abramson

January 1977 **Proceedings of the 1977 annual conference**

Full text available:  pdf(509.48 KB)


Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We consider macro processors to be a particular kind of translator writing system rather than simply a mechanism for performing textual replacement. This consideration leads to the application of modern compiler and translator writing system technology to the design and implementation of general macro processors. REGULUS is a general purpose macro processor based on regular expressions and is designed following the above mentioned considerations. REGULUS may be fitted to a variety of lexica ...

20 Common Lisp Object System specification

Daniel G. Bobrow, Linda G. DeMichiel, Richard P. Gabriel, Sonya E. Keene, Gregor Kiczales, David A. Moon

September 1988 **ACM SIGPLAN Notices**, Volume 23 Issue SI

Full text available:  pdf(6.88 MB)

Additional Information: [full citation](#), [citations](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

IEEE Xplore[®]
RELEASE 1.8Welcome
United States Patent and Trademark OfficeIEEE Xp
1 Million D
1 Million U

» Search Re

Help FAQ Terms IEEE Peer Review

Quick Links

Welcome to IEEE Xplore[®]

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **5** of **1049776** documents.A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or entering a new one in the text box.

string<and>macro

Search

☐ Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard1 **Hybrid genetic algorithms for constrained placement problems***Schnecke, V.; Vornberger, O.;*

Evolutionary Computation, IEEE Transactions on , Volume: 1 , Issue: 4 , Nov. 1997

Pages:266 - 277

[\[Abstract\]](#) [\[PDF Full-Text \(280 KB\)\]](#) **IEEE JNL**2 **An instrument control and data analysis program configured for NMR imaging***Roos, M.S.; Mushlin, R.A.; Veklerov, E.; Port, J.D.; Ladd, C.; Harrison, C.G.;*

Nuclear Science, IEEE Transactions on , Volume: 36 , Issue: 1 , Feb. 1989

Pages:988 - 992

[\[Abstract\]](#) [\[PDF Full-Text \(476 KB\)\]](#) **IEEE JNL**3 **CPP denotational semantics***Favre, J.-M.;*

Source Code Analysis and Manipulation, 2003. Proceedings. Third IEEE International Workshop on , 26-27 Sept. 2003

Pages:22 - 31

[\[Abstract\]](#) [\[PDF Full-Text \(405 KB\)\]](#) **IEEE CNF**4 **Task division and tracking based on macro-micro manipulators system for discontinuous trajectory***Chen Qijun; Wang Yuejuan; Chen Huitang;*

Systems, Man and Cybernetics, 2002 IEEE International Conference on , Volume: 4 , 6-9 Oct. 2002

Pages:6 pp. vol.4

[\[Abstract\]](#) [\[PDF Full-Text \(502 KB\)\]](#) IEEE CNF

5 Splicing systems on graphs

Freund, R.;

Intelligence in Neural and Biological Systems, 1995. INBS'95, Proceedings., First International Symposium on , 29-31 May 1995

Pages:189 - 194

[\[Abstract\]](#) [\[PDF Full-Text \(408 KB\)\]](#) IEEE CNF

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved



search string macro transform parsing










Search

Shortcuts Advanced Search Preferences

Web - (What's new?) Results **1 - 20** of about **17,500** for **string macro transform parsing**. Search took 0.17 seconds. (About this page...)

1. [btparse - C library for parsing and processing BibTeX data files](#)
... case (char **transform**, char * **string**, ushort options); DESCRIPTION. btparse is a C library for **parsing** and pro
definition" (for **string** entries), "preamble ...
starship.python.net/~gward/btOOL/doc/btparse.html - 23k - [Cached](#)
2. [The GNU C Library](#)
... Header Files. **Macro** Definitions of Functions ... Output Functions. **Parsing** a Template **String**. Example of Par
www.cee.hw.ac.uk/~ceebw1/libc.htm - 525k - [Cached](#)
3. [The GNU C Library](#)
... Header Files. **Macro** Definitions of Functions ... Output Functions. **Parsing** a Template **String**. Example of Par
www.astro.uni-bonn.de/~webstw/cm/gnu/libc.html - 525k - [Cached](#)
4. [Text::BibTeX - interface to read and parse BibTeX files](#)
... shutdown functions. Generic **string**-processing functions. Entry-**parsing** functions. **Macro** table functions ... Tra
TRANSFORM (a single character, one ...
theoryx5.uwinnipeg.ca/CPAN/data/Text-BibTeX/BibTeX.html - 33k - [Cached](#)
5. http://www.nofadz.com/moses/unix_man/cp.txt
... the C compiler to **transform** your program before actual ... nearly all other **parsing** and before preprocessing d
arguments converted into **string** constants ...
www.nofadz.com/moses/unix_man/cp.txt - 109k - [Cached](#)
6. [The C Preprocessor](#)
The C Preprocessor. Last revised July 1992. for GCC version 2. Richard M. Stallman. The C Preprocessor. The C
used automatically by the C compiler to **transform** your program before actual compilation. ... nearly all other par
transforming one **macro** argument into one **string** constant: there ...
www.chemie.fu-berlin.de/chemnet/use/info/cpp/cpp.html - 118k - [Cached](#)
7. [The C Preprocessor - The C Preprocessor](#)
The C Preprocessor. The C preprocessor is a **macro** processor that is used automatically by the C compiler to tr
compilation. ... before nearly all other **parsing** and before preprocessing directives ... This **macro** expands to a st
www.cs.ubc.ca/local/software/GNU_info/cpp_1.html - 115k - [Cached](#)
8. [The C Preprocessor - The C Preprocessor](#)
The C Preprocessor. The C preprocessor is a **macro** processor that is used automatically by the C compiler to tr
compilation. ... before nearly all other **parsing** and before preprocessing directives ... This **macro** expands to a st
www.mimas.ac.uk/utildocs/cpp_1.html - 111k - [Cached](#)
9. [Scheme Programming](#)
... CGI:url-unquote: a CGI script utility for **parsing** of a QUERY_**STRING** and POST messages ... map set! CPS t
syntax-rules: **Macro**-expander as a ...
www.pobox.com/~oleg/ftp/Scheme - 37k - [Cached](#)
10. [The C Preprocessor](#)
The C Preprocessor. Last revised March 1997. for GCC version 2. Richard M. Stallman. The C Preprocessor. The

is used automatically by the C compiler to **transform** your program before actual compilation. ... before nearly all directives ... This **macro** expands to a **string** (not a **string** constant ...
www.brightstareng.com/tool/doc/cpp.html - 120k - [Cached](#)

11. [ACM Lisp Pointers 4, 2 \(Apr/Jun 1991\), 3-15.](#) 
... only parse, but also to **transform**. In this way ... **parsing** of format control strings, in which major portions of the Our matchit **macro** ...
home.pipeline.com/~hbaker1/Prag-Parse.html - 51k - [Cached](#)
12. [The C Preprocessor: 1. The C Preprocessor](#) 
... the C compiler to **transform** your program before actual ... nearly all other **parsing** and before preprocessing d argument into one **string** constant: there ...
gcc.gnu.org/onlinedocs/gcc-2.95.3/cpp_1.html - 175k - [Cached](#)
13. <http://www.veillard.com/XML/XSLT/libxslt-api.xml> 
Will return from the function. </info> </macro> <macro name='CHECK_STOPPED0' file='xsltInternals'> <info>Ma
be stopped. ... file> <file name='transform'> <exports symbol='xslHandleDebugger ... vendor" **string** for this proc
error messages while not **parsing** or validating ...
www.veillard.com/XML/XSLT/libxslt-api.xml - 140k - [Cached](#)
14. [The GNU C Library](#) 
... and conversion functions. **String** and Array Utilities: Utilities for ... in your programs. **Macro** Definitions: Some f
Template **String**: What kinds of ...
www.techfak.uni-bielefeld.de/~clange/libc.html - 524k - [Cached](#)
15. [libutils.html](#) 
Libutils is a library of useful routines. SYNOPSIS. include <util.h> #include <dstring.h> #include <lists.h> cc ... -lut
of commonly-used routines which fall into the following classes:
tensile.sunsite.dk/libutils/libutils.html - 42k - [Cached](#)
16. <http://www.prakinf.tu-ilmenau.de/~czarn/ECOOP02GPW/nizhegorodov.ppt> (MICROSOFT POWER
... Closure **parsing** a template body from an input stream. **String** "x + y ... Syntax-to-syntax transforms. **MACRO**:
code generator ...
www.prakinf.tu-ilmenau.de/~czarn/ECOOP02GPW/nizhegorodov.ppt - 74k - [View as html](#)
17. <http://xmlsoft.org/XSLT/libxslt-api.xml> 
They are not part of the API or ABI, i.e. they can change without prior notice, use carefully. </
xmlsoft.org/XSLT/libxslt-api.xml - 160k - [Cached](#)
18. [PApp::XML - pxml sections and more](#) 
... called during **parsing** and the resulting **string** will be ... phtml></macro> As you can see, it uses XSLT's trans
DOM translation ...
www.goof.com/pcg/marc/man/XML.html - 20k - [Cached](#)
19. [Hundreds of Free Sample SPSS Syntax for Download](#) 
Archive of 400+ sample SPSS syntax, **macro** and scripts classified by purpose. ... not familiar with macros, see th
results.SPS. **Parsing** a variable which has embedded line ... to number.sps. **Transform string** coding into numbe
pages.infinet.net/rlevesqu/SampleSyntax.htm - 165k - [Cached](#)
20. [POV-Ray Quick Reference](#) 
POV-Ray Scene Description Language Quick Reference. Derived from the POV-Ray 3.1g User Documentation. b
and refinement of the entire syntax for the POV-Ray™ Scene Description Language.
www.donovansweb.com/~chaffe/quickref31g.html - 64k - [Cached](#)

Results Page:

1 2 3 4 5 6 7 8 9 10 ► [Next](#)

[Web](#) | [Images](#) | [Directory](#) | [Yellow Pages](#) | [News](#) | [Products](#)

Your Search:

Help us improve your search experience. [Send us feedback.](#)

Yahoo! Search is hiring! [Learn about job opportunities](#)

Copyright © 2004 Yahoo! Inc. All rights reserved. [Privacy Policy](#) - [Terms of Service](#) - [Submit Your Site](#)



Web Images Groups News Froogle [more »](#)

string macro transform parsing

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 7,510 for **string macro transform parsing**. (0.48 seconds)

[btparse](#) - C library for **parsing** and processing BibTeX data files

... ushort options); void bt_change_case (char **transform**, char * **string** ... implemented is that field and **macro** names may ... Some names in the above example: **string** , and ...

www.gerg.ca/software/btOOL/doc/btparse.html - 24k - [Cached](#) - [Similar pages](#)

[Text::BibTeX](#) - interface to read and parse BibTeX files

... OPTIONS)) Transforms the case of **STRING** according to **TRANSFORM** (a single ... that **STRING** is not modified in-place—the input **string** is copied ... **Macro** table functions. ...

www.gerg.ca/software/btOOL/doc/Text/BibTeX.html - 26k - [Cached](#) - [Similar pages](#)

[ACM Lisp Pointers](#) 4, 2 (Apr/Jun 1991), 3-15.

... about 222usec/char., and read-from-**string** took about ... ability of a Scheme compiler to **transform** the nest ... technique to produce a "reader **macro**" which translates ...

home.pipeline.com/~hbaker1/Prag-Parse.html - 51k - [Cached](#) - [Similar pages](#)

[The GNU C Library](#) - **String** and Array Utilities

... This function is implemented only as a **macro** which means one cannot get ... transformed = (char *) xmalloc (length); /* **Transform** array[i] ... Finding Tokens in a **String**. ...

www.ia.pw.edu.pl/~wujek/dokumentacja/gnu/libc/libc_5.html - 48k - [Cached](#) - [Similar pages](#)

[The GNU C Library](#) - **String** and Array Utilities

... a cleaner program to hide them behind a **macro** definition, like ... array[i]) * 2; temp_array[i].input = array[i]; /* **Transform** array[i] ... Finding Tokens in a **String**. ...

www.cs.utah.edu/dept/old/texinfo/glibc-manual-0.02/library_5.html - 39k - [Cached](#) - [Similar pages](#)

[POV-Ray: Documentation: 2.7.4 Language Directives](#)

... VECTOR | COLOR | USER_FUNCTION | **STRING** | ARRAY_REFERENCE | SPLINE | **TRANSFORM** | CAMERA | LIGHT ... debug **STRING** | #error **STRING** | #warning **STRING** ... 2.7.4.8 Macro. ...

www.povray.org/documentation/view/3.6.0/108/ - 18k - Jul 9, 2004 - [Cached](#) - [Similar pages](#)

[\[PDF\] The metafront System: Extensible **Parsing** and Transformation](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... Regarding **macro** processors, we refer to the ... forces AndExpressionRest to consume only the empty **string**. ... and output languages and **transform** syntactically legal ...

www1.elsevier.com/gej-ng/31/29/23/133/52/31/82.3.009.pdf - [Similar pages](#)

[An example of an interesting \(S\)XML transformation ; ; The example ...](#)

... is the sample document to **transform**, in its ... sort of a 'destructuring-bind' (define-macro (match-bind ... lambda (var) `(var (cdr (assq ' (**string**->symbol (**string** ...

okmij.org/ftp/Scheme/pull-punct-sxml.scm - 10k - [Cached](#) - [Similar pages](#)

[Round Tables](#)

... Using the following **macro** #define fe(v,it) for ... of "priority_queue", "pss" instead of "pair<**string**,**string**>", "vii" instead of ... in s to upper case **transform**(s.begin ...

www.topcoder.com/rtables/viewThread.jsp?&forum=327735&thread=330765&mc=9 - 77k - [Cached](#) - [Similar pages](#)

[\[PDF\] POV-Ray Scene Description Language Quick Reference](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... Directives Control the **parsing** of sections of ... **STRING** | **ARRAY_REFERENCE** | **SPLINE** |
TRANSFORM | **CAMERA** | **LIGHT** ... error **STRING** | #warning **STRING** Macro Define a macro ...
www.deakin.edu.au/~agoodman/scc308/povref.pdf - [Similar pages](#)

Google 

Result Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google